Event biasing I

Overview and Geometrical importance biasing

Toyama National College of Maritime Tech.
Tsukasa Aso

Geant4 Tutorial Course

Contents

- Overview
- Geometric biasing
- Setup of Geometric biasing
- Examples/extended/biasing
- Plan

Event biasing Overview

Event biasing in Geant4

- Event biasing (variance reduction) techniques are a vital requirement for many applications.
- Since Geant4 is a toolkit and also all source code is open, the user can do whatever he/she wants.
  - Capable users in experiments/institutions created their own implementations of event biasing for their needs.
- These feature could be utilized by many application fields such as
  - Shielding
  - Radiation environment assessment
  - Dosimetry
- It is more convenient for the user if Geant4 itself provides most commonly used event biasing techniques.

Event biasing techniques

- Production cuts / threshold
  - This is a biasing technique – most popular for many applications
- Geometry based biasing
  - Importance weighting for volume/region
  - Duplication or sudden death of tracks
- Leading particle biasing
  - Taking only the most energetic (or most important) secondary
- Primary event biasing
  - Biasing primary events and/or primary particles in terms of type of event, momentum distribution, etc.
- Forced interaction
  - Force a particular interaction, e.g. within a volume
- Enhanced process or channel
  - Increasing cross section for a process
- Physics based biasing
  - Biasing secondary production in terms of particle type, momentum distribution, cross-section, etc.

=> Weight on track / event.

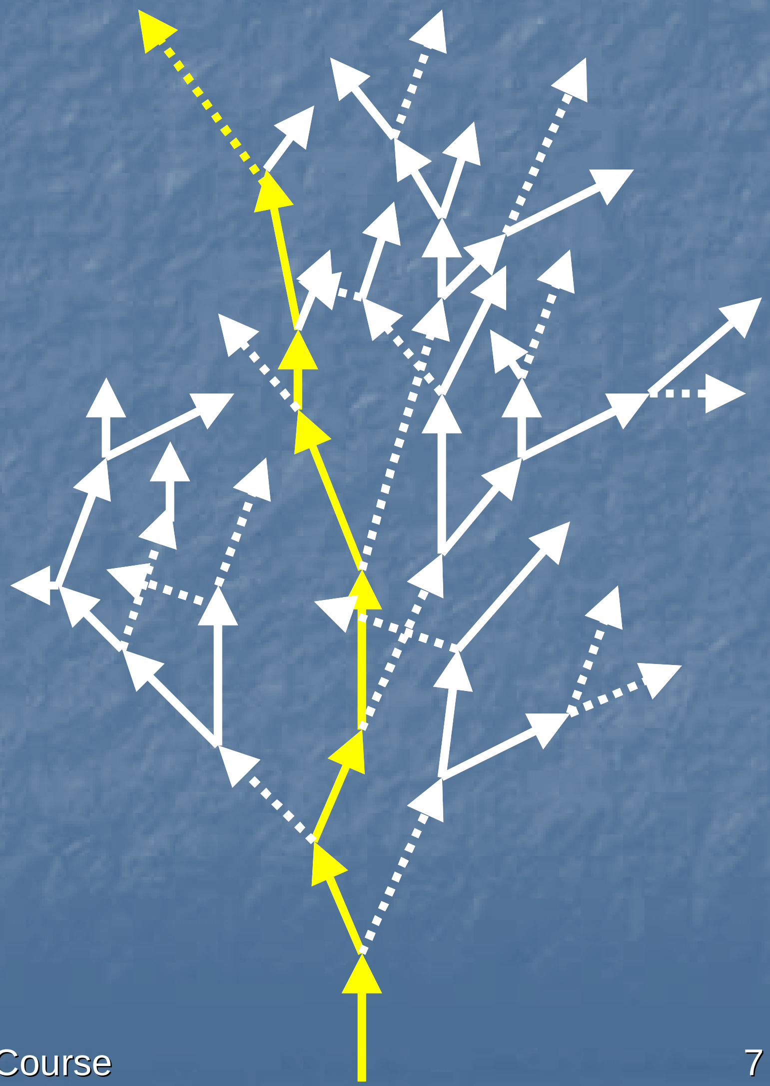
Current features in Geant4

- Partial MARS migration
  - n, p, pi, K (< 5 GeV)
    - Since Geant4 0.0
- General particle source module
  - Primary particle biasing
    - Since Geant4 3.0
- Radioactive decay module
  - Physics process biasing in terms of decay products and momentum distribution
    - Since Geant4 3.0
- Geometry based biasing
  - Weight associating with real volume or artificial volume
    - Since Geant4 4.2
  - Weight cutoff and weight window
    - Since Geant4 5.2
- Hadronic process module
  - Cross-section biasing (PhotoInelactic,ElectronNuclear,PositronNuclear)
  - Leading particle biasing for hadronic processes
    - Since Geant4 7.0

Leading particle biasing

- Estimation of penetrating particles -

- Simulating a full shower is an expensive calculation.
- Instead of generating a full shower, trace only the most energetic secondary.
  - Other secondary particles are immediately killed before being stacked.
  - Convenient way to roughly estimate, e.g. the thickness of a shield.
  - Of course, physical quantities such as energy are not conserved for each event.



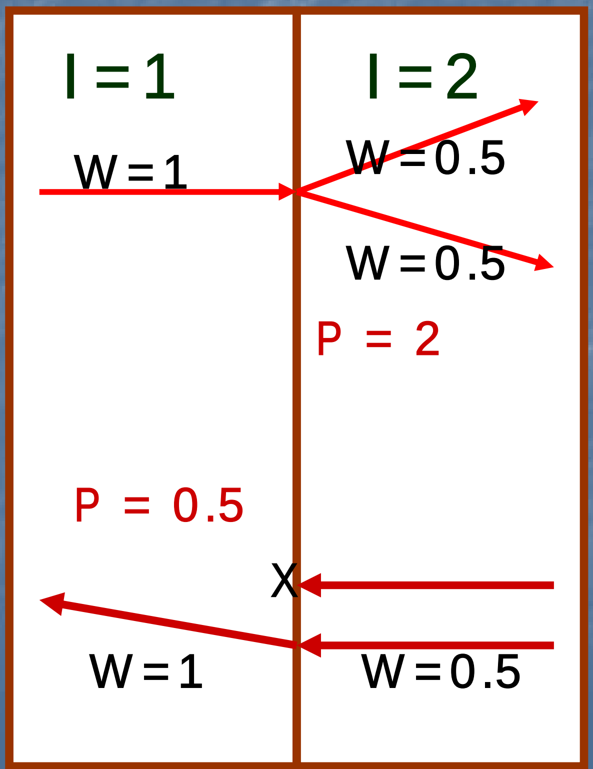
Geometric Biasing

The purpose of geometry based event biasing is to save computing time by sampling less often the particle histories entering “less important” geometry regions, and more often in more “important” regions.

* Importance sampling technique
* Weight window technique

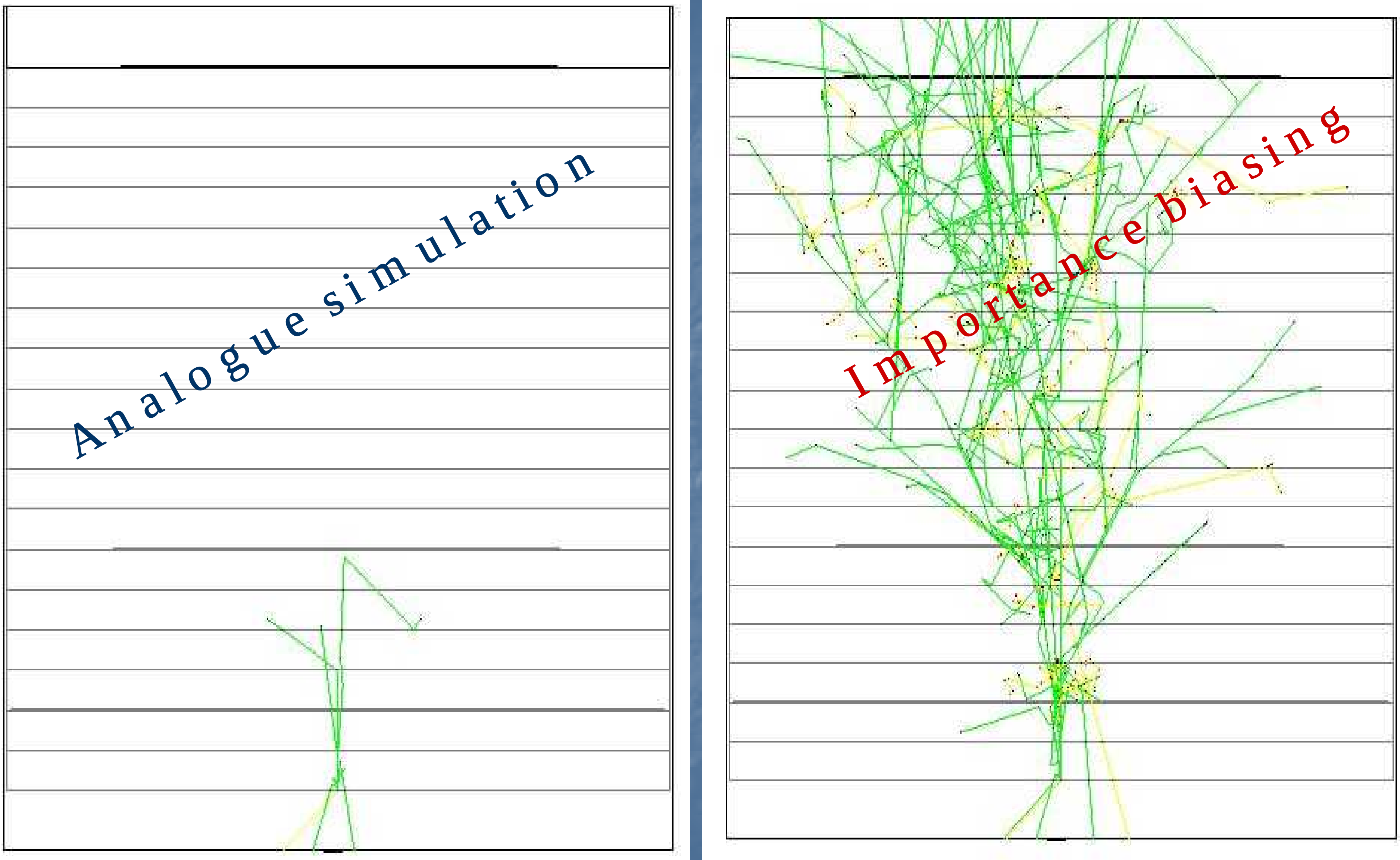
Importance sampling technique

- Importance sampling acts on particles crossing boundaries between "importance cells".
- The action taken depends on the importance value assigned to the cell.
- In general, a track is played either split or Russian roulette at the geometrical boundary depending on the importance value assigned to the cell.



- Survival probability (P) is defined by the ratio of importance value.
  $P = I_{post} / I_{pre}$
- The track weight is changed to W/P.
- Splitting a track (P > 1)
  - E.g. creating two particles with half the 'weight' if it moves into volume with double importance value.
- Russian-roulette (P < 1) in opposite direction
  - E.g. Kill particles according to the survival probability (1 - P).

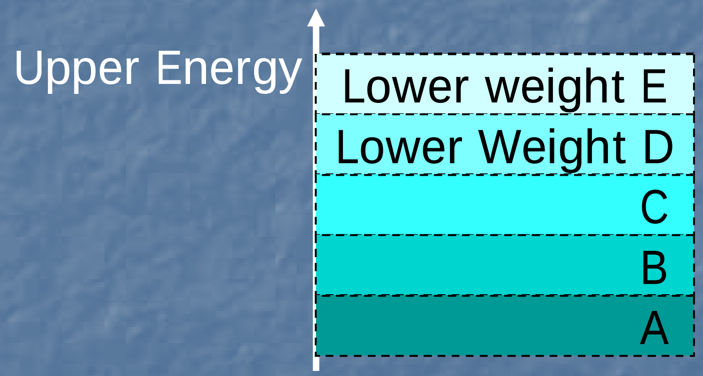
Importance biasing

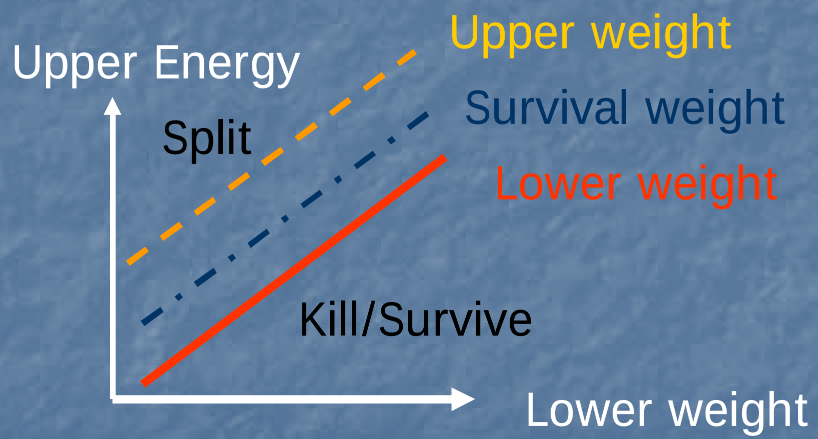


10 MeV neutron in thick concrete cylinder

The Weight Window Technique

- The weight window technique is a weight-based algorithm – generally used together with other techniques as an alternative to importance sampling:
  - It applies splitting and Russian roulette depending on space (cells) and energy
  - User defines weight windows in contrast to defining importance values as in importance sampling
- A weight window may be specified for every cell and for several energy regions: *space-energy cell* .

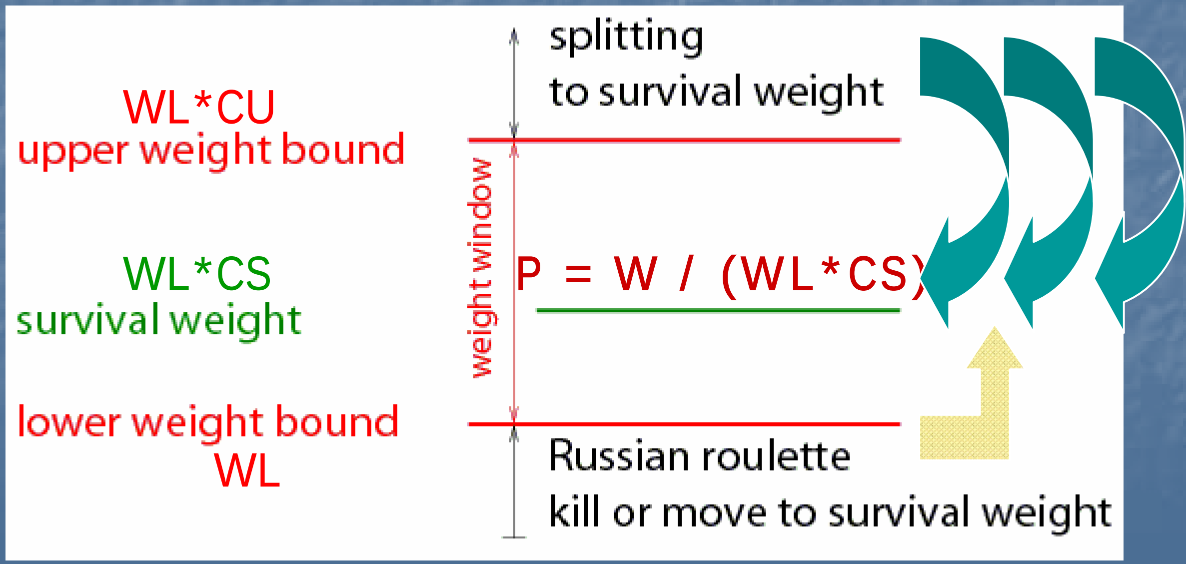




- Apply in combination with other techniques such as cross-section biasing, leading particle and implicit capture, or combinations of these.

The weight window technique (continue)

- Checks the particle weight
  - Compare the particle weight with a 'window' of weights defined for the current ***energy-space*** cell
  - Play splitting or roulette in case if it is outside, resulting in 0 or more particles 'inside' the window
    - E.g. WL is a lower weight bound of a cell.
      CU and CS are upper limit factor and survival factor, respectively.
      - W > WL*CU Split track
      - W < WL*WL Roulette

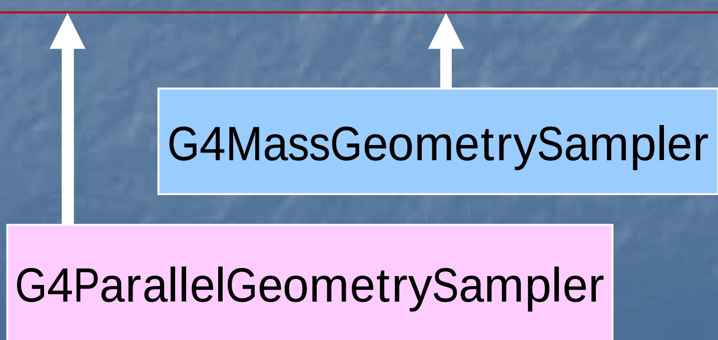


Setup of geometric biasing

Sampler for Geometry based biasing

- Sampler is the top level class to apply importance sampling and weight roulette.

```
class G4VSampler{
 public:
   virtual  ~G4VSampler(){}
   virtual  void PrepareScoring(G4VScorer* scorer) =0;
   virtual  void PrepareImportanceSampling(G4VIStore* istore,
                                                   const G4VImportanceAlgorith* iag=0)=0;
   virtual  void PrepareWeightWindow(G4VWeightWindowStore *wwstore,
                                              G4VWeightWindowAlgorithm *wwAlg=0,
                                              G4PlaceOfAction placeOfAction= onBoundary)=0;
   virtual  void PreareWeightRoulette(G4double wsurvive=0.5,G4double wlimit=0.25,
                                              G4double isource=1) = 0;
   virtual void Configure() = 0;
   .... Snipped ....
}
```



Two concrete classes are provided for the "mass" and the "parallel" geometry, respectively.

Concrete sampler classes

- A concrete sampler class is responsible for one particle type which should be given to the constructor of the sampler class
- Depending on the biasing technique, the Configure() method of a sampler will setup specialized processes, e.g. transportation in the parallel geometry, importance sampling, weight window, etc. for the given particle type.

- G4MassGeometrySampler
  - For a geometry of the experiment to be simulated.
- G4ParallelGeometrySampler
  - For a “parallel” geometry, “importance geometry”, which may be constructed to define physical volumes according to which variance reduction are applied.
  - Restriction/Limitation
    - World of "importance geometry" must 'overlap' exactly with "mass" world
    - Biasing and scoring of charged particles in a field is not yet supported when using an "importance geometry".

Setup importance biasing

- Decide whether to bias in "mass" geometry,
  or in a dedicated 'parallel' geometry ("importance geometry")
- Assign an importance value to a volume for all of volumes in this geometry
  - The importance is a (double) number.
  - G4IStore class is used to store importance values and related geometry cells.

```
Class G4IStore : public G4VIStore{
 public:
   explicit G4IStore(const G4VPhysicalVolume &worldvolume);
   void AddImportanceGeometryCell(G4double importance,  const G4GeometryCell &gCell);
   void AddImportanceGeometryCell(G4double importance,  const G4VPhysicalVolume &,
                                                        G4int aRepNum = 0);
   --- snipped ---
```

- Register the processes for importance biasing for each particle type
  - G4MassGeometrySampler / G4ParallelGeometrySampler prepares the process. The particle type is given to the constructor.
    - This configuration is prepared by PrepareImportanceSampling(G4VISotore*, G4VImportanceAlgorithm*); with importance values and the algorithm.
      - The default algorithm is implemented in G4ImportanceAlgorithm class derived from G4VImportanceAlgorithm base class.

Setup weight window biasing

- Assign a lower weight bound value and a UpperEnergy bound to a volume for all of volumes in this geometry
  - Both bounds are given in (double) numbers.
  - G4WeightWindowStore class stores bound values in cells.

```
Class G4WeightWindowStore : public G4VWieghtWindowStore{
 public:
  explicit G4WeightWindowStore(const G4VPhysicalVolume &worldvolume);
  void AddUpperEboundLowerWeightPairs(const G4GeometryCell &gCell,
                                      const G4UpperEnergyToLowerWeightMap& enWeMap);
  void AddLowerWeights(const G4GeometryCell &gCell,
                       const std::vector<G4double> &lowerWeights);
  void SetGeneralUpperEnergyBounds(const
                       std::set<G4double,std::less<G4double>> &enBounds);
--- snipped ---
```

- Sampler prepares processes to a particle type.
  - PrepareWeightWindowStore(G4VWeightWindowStore*, G4VWeightWindowAlgorithm*, G4PlaceOfAction);

    Boundary, Collision, or Both
  - The weight window algorithm is provided by
    - G4WeightWindowAlgorithm(G4double upperLimitFaktor, G4double survivalFaktor, G4int maxNumberOfSplits);
      - upperLimitFaktor/suvivalFaktor are multiplied to a lower weight bound to define the upper weight and the survival weight.

More on using importance biasing

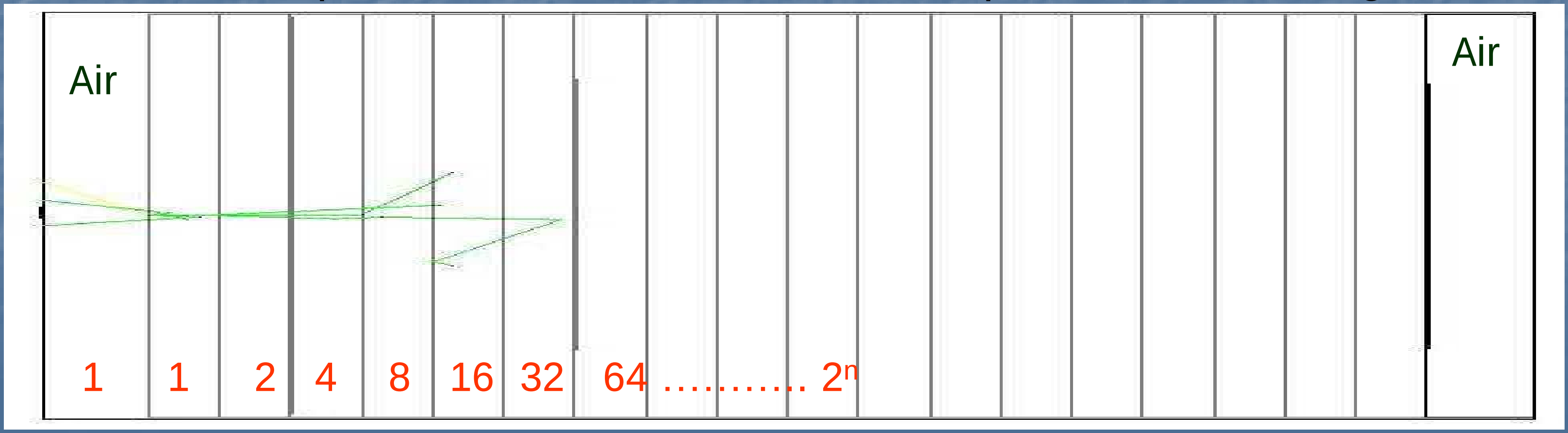
- Optional capability
  - Scorer may be used with the Sampler using PrepareScorer(G4Vscorer*).
    - G4Scorer is provided as a concrete scoring class.
    - A "scorer" class derives from the interface G4VScorer. Users may create customized "scorers". See exampleB02 for detail.

  - Checking importance sampling
    - A default implementation is provided through *G4Scorer*, which provides following parameters:
      - → Importance
      - Tracks entering : Number of tracks entering the cell
      - Population : Number of tracks including produced in the cell
      - Collisions : Number of steps limited by physics process in the cell
      - Coll*WGT : Weighted sum of collisions
      - NumWGTedE : Number weighted energy
      - FluxWGTedE : Flux weighted energy
      - Av. Track WGT : Average track weight

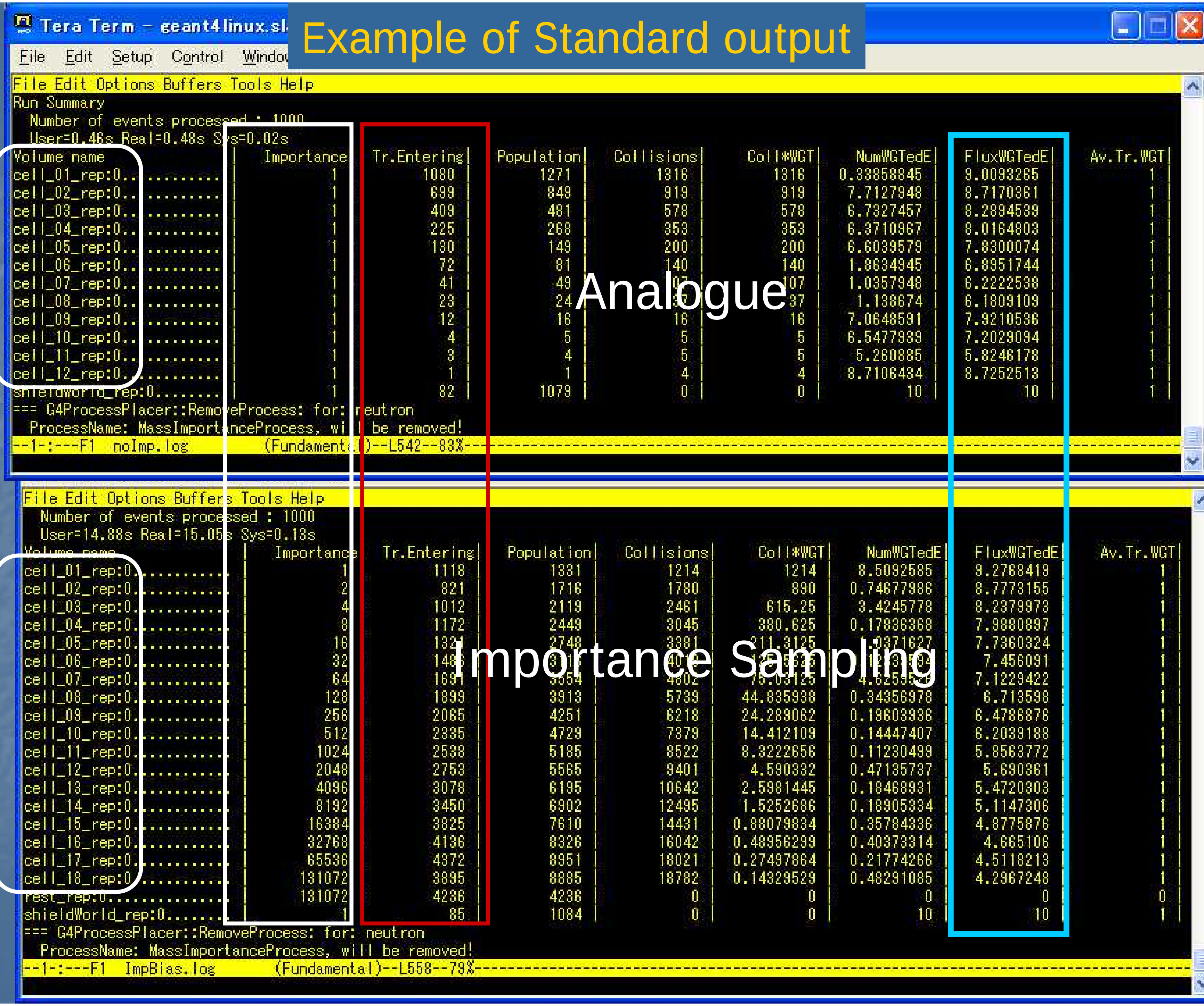
Examples/extended/biasing

Biasing example B01

- Shows the importance sampling in the mass (tracking) geometry
- Option to show weight window
- 10 MeV neutron shielding by cylindrical thick concrete material
- Geometry
  - 80 cm high concrete cylinder divided into 18 slabs
  - Importance values assigned to 18 concrete slabs in the DetectorConstruction for simplicity.
  - The G4Scorer is used for the checking result
    - Top level class uses the framework provided for scoring.



Example of Standard output



File Edit Options Buffers Tools Help

Run Summary

Number of events processed : 1000

User=0.46s Real=0.48s Sys=0.02s

Volume name	Importance	Tr.Entering	Population	Collisions	Coll*WGT	NumWGTedE	FluxWGTedE	Av.Tr.WGT
cell_01_rep:0...........	1	1080	1271	1316	1316	0.33858845	9.0093265	1
cell_02_rep:0...........	1	699	849	919	919	7.7127948	8.7170361	1
cell_03_rep:0...........	1	409	481	578	578	6.7327457	8.2894539	1
cell_04_rep:0...........	1	225	268	353	353	6.3710967	8.0164803	1
cell_05_rep:0...........	1	130	149	200	200	6.6039579	7.8300074	1
cell_06_rep:0...........	1	72	81	140	140	1.8634945	6.8951744	1
cell_07_rep:0...........	1	41	49	[illegible]	107	1.0357948	6.2222538	1
cell_08_rep:0...........	1	23	24	[illegible]	37	1.138674	6.1809109	1
cell_09_rep:0...........	1	12	16	16	16	7.0648591	7.9210536	1
cell_10_rep:0...........	1	4	5	5	5	6.5477939	7.2029094	1
cell_11_rep:0...........	1	3	4	5	5	5.260885	5.8246178	1
cell_12_rep:0...........	1	1	1	4	4	8.7106434	8.7252513	1
shieldWorld_rep:0........	1	82	1079	0	0	10	10	1

=== G4ProcessPlacer::RemoveProcess: for: neutron

ProcessName: MassImportanceProcess, will be removed!

--1-:---F1 noImp.log (Fundamental)--L542--83%--

File Edit Options Buffers Tools Help

Number of events processed : 1000

User=14.88s Real=15.05s Sys=0.13s

Volume name	Importance	Tr.Entering	Population	Collisions	Coll*WGT	NumWGTedE	FluxWGTedE	Av.Tr.WGT
cell_01_rep:0...........	1	1118	1331	1214	1214	8.5092585	9.2768419	1
cell_02_rep:0...........	2	821	1716	1780	890	0.74677986	8.7773155	1
cell_03_rep:0...........	4	1012	2119	2461	615.25	3.4245778	8.2379973	1
cell_04_rep:0...........	8	1172	2449	3045	380.625	0.17836368	7.9880897	1
cell_05_rep:0...........	16	[illegible]	2748	3381	211.3125	[illegible]	7.7360324	1
cell_06_rep:0...........	32	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	7.456091	1
cell_07_rep:0...........	64	1691	[illegible]	[illegible]	75.03125	[illegible]	7.1229422	1
cell_08_rep:0...........	128	1899	3913	5739	44.835938	0.34356978	6.713598	1
cell_09_rep:0...........	256	2065	4251	6218	24.289062	0.19603936	6.4786876	1
cell_10_rep:0...........	512	2335	4729	7379	14.412109	0.14447407	6.2039188	1
cell_11_rep:0...........	1024	2538	5185	8522	8.3222656	0.11230499	5.8563772	1
cell_12_rep:0...........	2048	2753	5565	9401	4.590332	0.47135737	5.690361	1
cell_13_rep:0...........	4096	3078	6195	10642	2.5981445	0.18468931	5.4720303	1
cell_14_rep:0...........	8192	3450	6902	12495	1.5252686	0.18905334	5.1147306	1
cell_15_rep:0...........	16384	3825	7610	14431	0.88079834	0.35784336	4.8775876	1
cell_16_rep:0...........	32768	4136	8326	16042	0.48956299	0.40373314	4.665106	1
cell_17_rep:0...........	65536	4372	8951	18021	0.27497864	0.21774266	4.5118213	1
cell_18_rep:0...........	131072	3895	8885	18782	0.14329529	0.48291085	4.2967248	1
rest_rep:0..............	131072	4236	4236	0	0	0	0	0
shieldWorld_rep:0........	1	85	1084	0	0	10	10	1

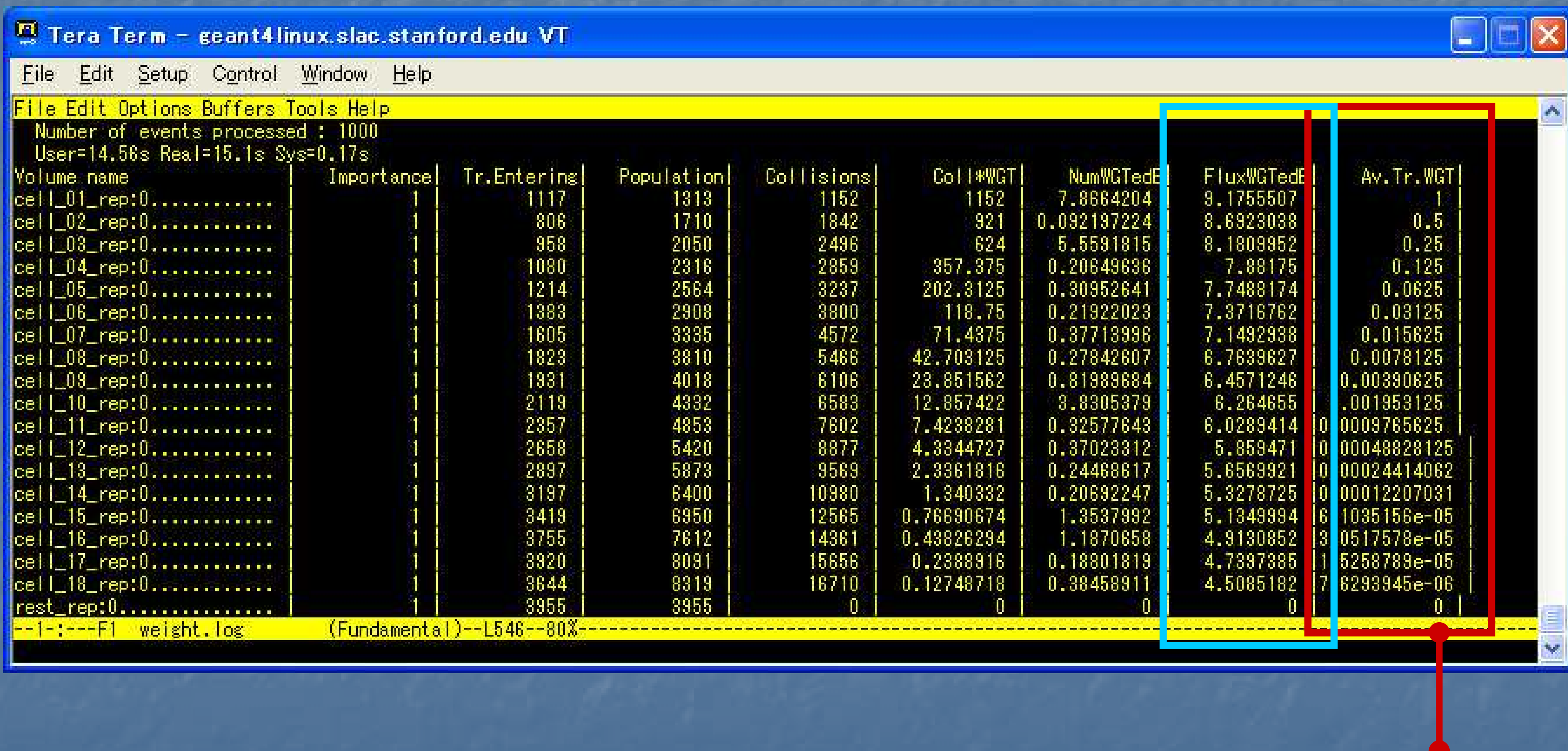
=== G4ProcessPlacer::RemoveProcess: for: neutron

ProcessName: MassImportanceProcess, will be removed!

--1-:---F1 ImpBias.log (Fundamental)--L558--79%--

Weight Window Technique

- Energy bounds < 1 GeV
- Upper limit factor : CU = 1, Survival factor : CS = 1
- Lower limit weight is proportional to 2^{-n} along to slabs



Tera Term - geant4linux.slac.stanford.edu VT

File Edit Setup Control Window Help

File Edit Options Buffers Tools Help

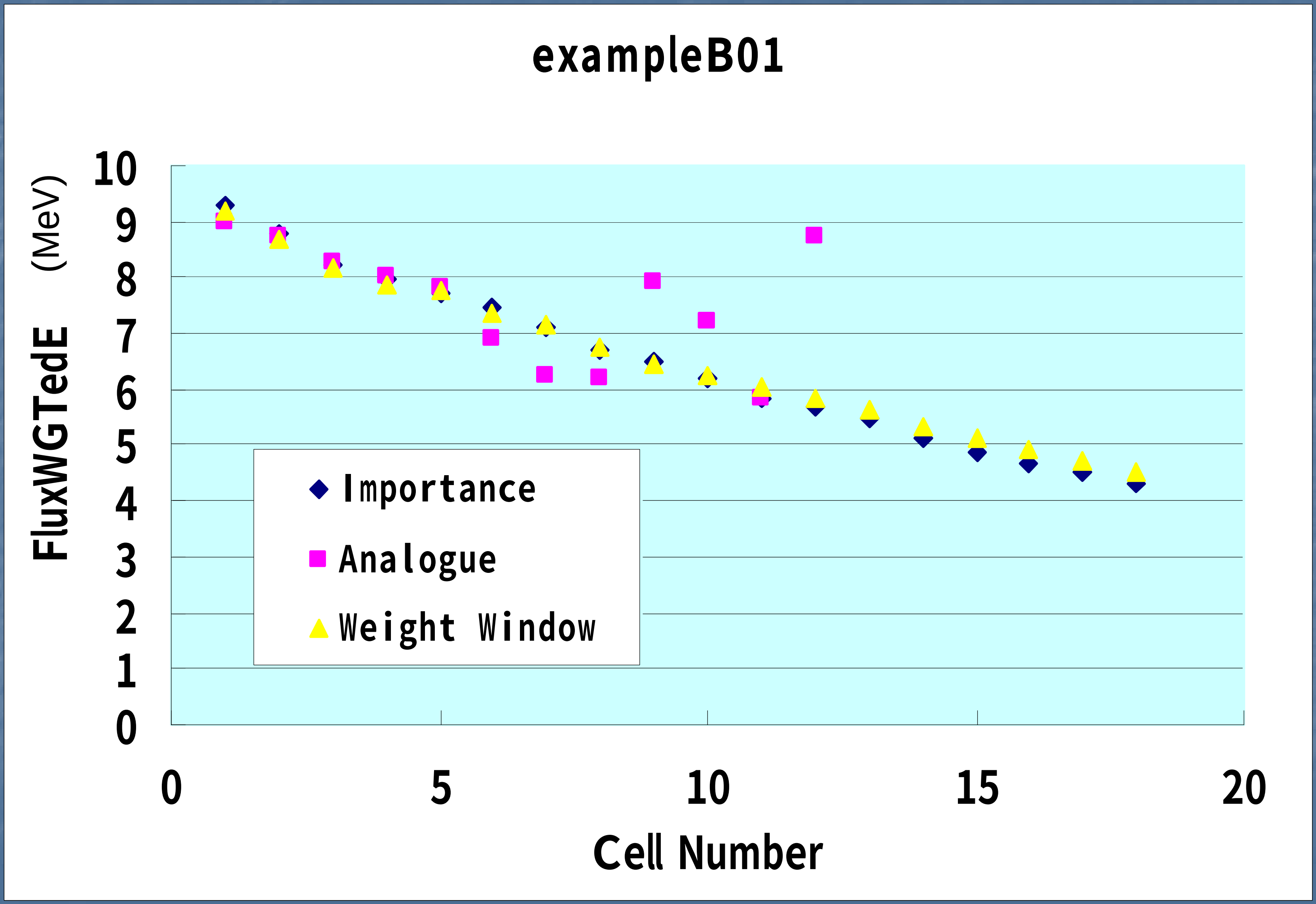
Number of events processed : 1000
User=14.56s Real=15.1s Sys=0.17s

Volume name	Importance	Tr.Entering	Population	Collisions	Coll*WGT	NumWGTedE	FluxWGTedE	Av.Tr.WGT
cell_01_rep:0............	1	1117	1313	1152	1152	7.8664204	9.1755507	1
cell_02_rep:0............	1	806	1710	1842	921	0.092197224	8.6923038	0.5
cell_03_rep:0............	1	958	2050	2496	624	5.5591815	8.1809952	0.25
cell_04_rep:0............	1	1080	2316	2859	357.375	0.20649636	7.88175	0.125
cell_05_rep:0............	1	1214	2564	3237	202.3125	0.30952641	7.7488174	0.0625
cell_06_rep:0............	1	1383	2908	3800	118.75	0.21922023	7.3716762	0.03125
cell_07_rep:0............	1	1605	3335	4572	71.4375	0.37713996	7.1492938	0.015625
cell_08_rep:0............	1	1823	3810	5466	42.703125	0.27842607	6.7639627	0.0078125
cell_09_rep:0............	1	1931	4018	6106	23.851562	0.81989684	6.4571246	0.00390625
cell_10_rep:0............	1	2119	4332	6583	12.857422	3.8305379	6.264655	.001953125
cell_11_rep:0............	1	2357	4853	7602	7.4238281	0.32577643	6.0289414	0 0009765625
cell_12_rep:0............	1	2658	5420	8877	4.3344727	0.37023312	5.859471	0 00048828125
cell_13_rep:0............	1	2897	5873	9569	2.3361816	0.24468617	5.6569921	0 00024414062
cell_14_rep:0............	1	3197	6400	10980	1.340332	0.20692247	5.3278725	0 00012207031
cell_15_rep:0............	1	3419	6950	12565	0.76690674	1.3537992	5.1349994	6 1035156e-05
cell_16_rep:0............	1	3755	7612	14361	0.43826294	1.1870658	4.9130852	3 0517578e-05
cell_17_rep:0............	1	3920	8091	15656	0.2388916	0.18801819	4.7397385	1 5258789e-05
cell_18_rep:0............	1	3644	8319	16710	0.12748718	0.38458911	4.5085182	7 6293945e-06
rest_rep:0...............	1	3955	3955	0	0	0	0	0

--1-:---F1 weight.log (Fundamental)--L546--80%----------------

Survival weight bounds

Flux multiplied by Kinetic energy of particle



Example B02

- B02 example for showing
  - importance sampling in a parallel geometry
  - a customized scoring making use of the scoring framework.
  - Mass geometry consists of a 180 cm high simple bulk concrete cylinder
  - A parallel geometry is created to hold importance values for slabs of width 10cm and for scoring.
    - Note: The parallel world volume must overlap the mass world volume
    - The radii of the slabs is larger than the radius of the concrete cylinder in the mass geometry.
    - The importance value is assigned to each 'G4GeometryCell'
      - Pairs of G4GeometryCell and importance values are stored in the importance store, G4IStore.
  - The scoring uses the G4CellSCorer and one customized scorer for the last slab.
  - It can be built and run using the PI implementation of AIDA
    - For this see http://cern.ch/PI.
  - At the end a histogram called "b02.hbook" is created.

Example B03

- Uses Geant4 importance sampling and scoring through python.
- It creates a simple histogram.
- It demonstrates how to use a customized scorer and importance sampling in combination with a scripting language, python.
- Geant4 code is executed from a python session.
  - Note: the swig package is used to create python shadow classes and to generate the code necessary to use the Geant4 libraries from a python session.
- It can be built and run using the PI implementation of AIDA
  - For this see http://cern.ch/PI.
- At the end a histogram called "trackentering.hbook" is created.

Plans for event biasing in Geant4

- Extend “importance” geometries for use with charged particles in field.
  - New navigation system for parallel geometries are under preparation and will be released soon.
  - Implementation of geometric biasing in “importance” geometry will be migrated to use this new navigation.
- Scorer may be separated from event biasing.
  - Scorer will be merged with sensitive detector for simplicity.

- User’s contribution is welcome.